Classes abtraites

Concept

Une **classe abstraite** est une classe qui ne peut pas être instanciée directement, mais qui sert plutôt de **modèle** ou de **base** pour d'autres classes. Elle est conçue pour **être héritée** et **étendue** par d'autres classes qui peuvent en revanche être instanciées.

Dans une classe abstraite, on pourra trouver :

- Des **méthodes et attributs classiques**, comme on en a déjà l'habitude en POO ;
- Des **méthodes dites « abstraites »** qui fournissent uniquement leur **signature**. Charge aux classes qui en héritent de définir le code de ces méthodes.

Exemple :

A.java

```
// une classe abstraite A
public abstract class A
{
    // une méthode non abstraite (=implémentée) dans A
    public void maMethode() {
        ...
    }

    // une méthode abstraite de A
    public abstract void autreMethode (int nombre) ;
}
```

B.java

```
// une classe qui dérive la classe abstraite A
public class B extends A
{

    // implémentation de la méthode abstraite de A : obligatoire,
    // sinon B reste abstraite
    public void autreMethode (int param) {
        ...
    }
}

// instanciation d'une classe qui dérive une classe abstraite : OK
B b = new B (...) ;
```

Bénéfices

- On peut placer dans une **classe abstraite** toutes les **fonctionnalités** dont on souhaite disposer dans ses **classes dérivées**, sans pour autant définir leur **fonctionnement interne**;
- On peut utiliser des **classes abstraites** pour simplement empêcher qu'une classe ne soit **instanciée** parce que son existence en tant que classe est **uniquement technique** et ne repose pas sur le besoin de manipuler l'objet en tant que tel.

Règles de construction

- Dès qu'une classe comporte une ou plusieurs méthodes abstraites, elle est abstraite, même si on ne l'indique pas avec le mot-clé « abstract » dans sa définition.
- Une classe qui hérite d'une classe abstraite n'est pas tenue de redéfinir toutes les méthodes abstraites (elle peut n'en définir que certaines). Dans ce cas elle demeure elle-même abstraite.

Comment les utiliser

Définir la classe abstraite en utilisant le mot-clé "abstract" devant la déclaration de classe.

```
public abstract class MaClasseAbstraite {
    // Déclaration de variables et/ou de méthodes abstraites
}
```

=== Définir des variables et/ou des méthodes abstraites dans la classe abstraite. ===

```
public abstract class MaClasseAbstraite {
    protected int variableAbstraite;

    public abstract void methodeAbstraite();
}
```

Hériter de la classe abstraite dans une classe concrète en utilisant le mot-clé "extends".

```
public class MaClasseConcrete extends MaClasseAbstraite {
    // Implémentation des méthodes et des variables de la classe abstraite
}
```

Implémenter les méthodes abstraites dans la classe concrète en utilisant le mot-clé

"override".

```
public class MaClasseConcrete extends MaClasseAbstraite {
    protected int variableAbstraite;

    public void methodeAbstraite() {
        // Implémentation de la méthode abstraite
    }
}
```

=== Utiliser la classe concrète pour instancier des objets et appeler les méthodes héritées de la classe abstraite. ===

```
MaClasseConcrete objet = new MaClasseConcrete();
objet.methodeAbstraite();
```